OMEGA3P: MODELING NEXT GENERATION PARTICLE ACCELERATORS*

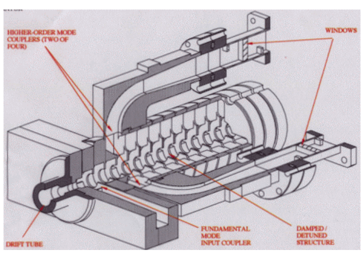
Brian McCandless, Zenghai Li, Yong Sun and Kwok Ko

Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309, USA

Abstract: The increasingly demanding design requirements of the next-generation particle accelerators such as the Next Linear Collider (NLC), have placed heavy emphasis on the accuracy and reliability of RF software in order that accelerator components can be modeled and analyzed with greater confidence. Presently popular codes are inefficient in handling complex geometric shapes, or are limited in their ability to solve large-scale problems. The Numerical Modeling Group (NMG) at SLAC has an ongoing effort to develop advanced numerical tools that specifically address these issues though the use of unstructured grids and multi processing capability. This poster presents Omega3P, a parallel distributed-memory finite-element code for solving electromagnetics in the frequency domain for large complex three-dimensional geometries. The problem is challenging because the distributed mesh operations are communication intensive, and the parallel eigensolver is computationally expensive. Omega3P has been successful in solving geometries consisting of millions of mesh points on the Cray T3E.

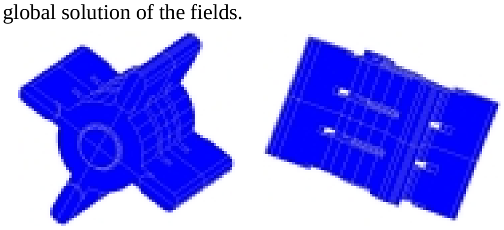
Introduction

PARALLEL COMPUTING motivated by increasing need to model LARGE, COMPLEX RF cavities ACCURATELY for Next-Generation Accelerators , such as the Damped, Detuned Structure (DDS) for the Next Linear Collider (NLC).



DDS Modeling

- Individual Cell Design - requires frequency accuracy to within 0.01 % because cell to cell dimensions vary on the same order.
- Whole Structure Analysis - simulates all 206 cells to obtain a global solution of the fields.



Both types of modeling are beyond desktop computing resources. SLAC is developing modeling tools that utilize multi-processors.

Omega3P - A Parallel Eigensolver

Based on Omega3, a finite element (linear & quadratic elements) eigensolver with mesh refinement capabilities.

Key Components for Parallelization

- Mesh Distribution - domain decomposition of the geometry mesh for balanced load on each processor.
- Matrix Assembly - finite element formulation and generation of the Mass and Stiffness for the generalized eigenvalue problem:

$$Kx = \lambda Mx$$

- Eigensovler - linear algebra operations to support the Jacobi-Davidson method.

Omega3P - Software Components

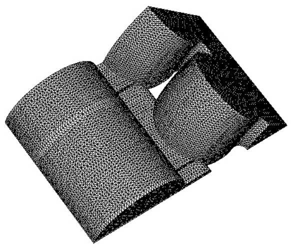
Omega3P - application module layered on DistMesh and EigenSolver and contains the finite element formulation and post processing. It uses MeshTV (from LLNL) for visualization.

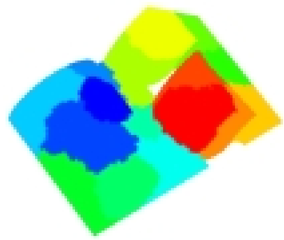
DistMesh - a library for operating on distributed unstructured meshes. Operations include parallel file I/O, partitioning, distribution, global numbering, and refinement. DistMesh makes use of ParMETIS (from U of Minnesota) for partitioning.

EigenSolver - code for solving the generalized eigenvalue problem using a Hybrid algorithms. It uses Aztec (from Sandia National Lab) for solving sparse linear systems in parallel.

* Work supported by the U.S. Department of Energy,
Division of Mathematical, Information and Computer Sciences,
and the Division of High Energy Physics.

DDS Cell - Domain Decomposition

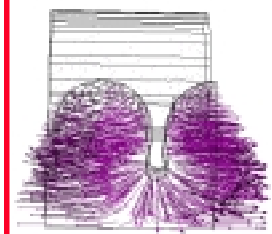


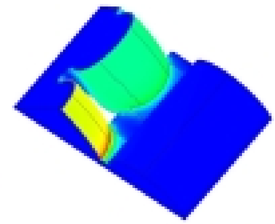


Mesh (with 462687 Elements) of one octant of 1.5 DDS Cells.

The same mesh partitioned into 16 (roughly equal) pieces

Post Processing with MeshTV





Above: Electric Fields
Above Right: Wall Loss
Right: Magnetic Fields

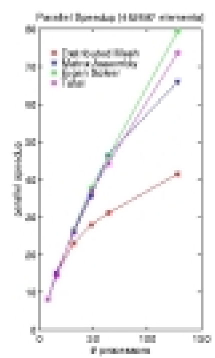
Hybrid Scheme in Eigensolver

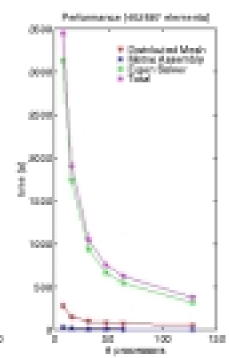
An optimized combination of three powerful methods.

- Spectrum transformation and band pass polynomial filtering,
- Inexact Krylov subspace projection,
- Modified Jacobi-Davidson local refinement.

This results in solver accuracy and convergence as well as parallel scalability far superior than other algorithms.

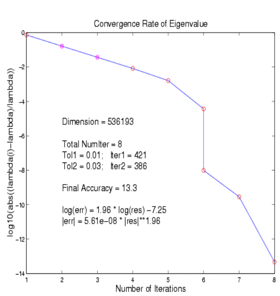
Performance

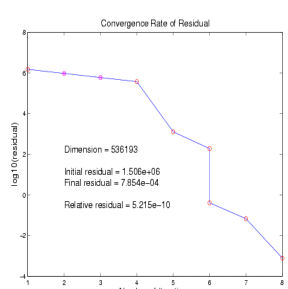




462687 Elements, 536193 Degrees of Freedom

Eigensolver Convergence Rate





An Inexact Krylov subspace projection algorithm is used for six steps, followed by two steps of the Jacobi-Davison Method.

Work in Progress

- Whole structure simulation (~100 Million Mesh Elements)
- Adaptive mesh refinement
- Parallel mesh generation
- Block version of the Hybrid Jacobi-Davidson eigensolver
- Port to other parallel platforms such as:
  - SGI Cray Origin
  - Sun Enterprise 10000
  - PC Cluster